

Notice of Allowability**Application No.**

10/591,317

Applicant(s)

KAZAKOV ET AL.

Examiner

CHIH-CHING CHOW

Art Unit

2191

- The MAILING DATE of this communication appears on the cover sheet with the correspondence address--

All claims being allowable, PROSECUTION ON THE MERITS IS (OR REMAINS) CLOSED in this application. If not included herewith (or previously mailed), a Notice of Allowance (PTOL-85) or other appropriate communication will be mailed in due course. **THIS NOTICE OF ALLOWABILITY IS NOT A GRANT OF PATENT RIGHTS.** This application is subject to withdrawal from issue at the initiative of the Office or upon petition by the applicant. See 37 CFR 1.313 and MPEP 1308.

1. ☒ This communication is responsive to 03/18/10.
2. ☒ The allowed claim(s) is/are 1-39.
3. ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some* c) ☐ None of the:
1. ☐ Certified copies of the priority documents have been received.
2. ☐ Certified copies of the priority documents have been received in Application No. ____.
3. ☐ Copies of the certified copies of the priority documents have been received in this national stage application from the International Bureau (PCT Rule 17.2(a)).

* Certified copies not received: ____.

Applicant has THREE MONTHS FROM THE "MAILING DATE" of this communication to file a reply complying with the requirements noted below. Failure to timely comply will result in ABANDONMENT of this application.

THIS THREE-MONTH PERIOD IS NOT EXTENDABLE

4. ☐ A SUBSTITUTE OATH OR DECLARATION must be submitted. Note the attached EXAMINER'S AMENDMENT or NOTICE OF INFORMAL PATENT APPLICATION (PTO-152) which gives reason(s) why the oath or declaration is deficient.
5. ☐ CORRECTED DRAWINGS (as "replacement sheets") must be submitted.
- (a) ☐ including changes required by the Notice of Draftsperson's Patent Drawing Review (PTO-948) attached
- 1) ☐ hereto or 2) ☐ to Paper No./Mail Date ____.
- (b) ☐ including changes required by the attached Examiner's Amendment / Comment or in the Office action of Paper No./Mail Date ____.
- Identifying Indicia such as the application number (see 37 CFR 1.84(c)) should be written on the drawings in the front (not the back) of each sheet. Replacement sheet(s) should be labeled as such in the header according to 37 CFR 1.121(d).
6. ☐ DEPOSIT OF and/or INFORMATION about the deposit of BIOLOGICAL MATERIAL must be submitted. Note the attached Examiner's comment regarding REQUIREMENT FOR THE DEPOSIT OF BIOLOGICAL MATERIAL.

Attachment(s)

1. ☒ Notice of References Cited (PTO-892)
2. ☐ Notice of Draftperson's Patent Drawing Review (PTO-948)
3. ☒ Information Disclosure Statements (PTO/SB/08),
Paper No./Mail Date 06/15/10
4. ☐ Examiner's Comment Regarding Requirement for Deposit of Biological Material
5. ☐ Notice of Informal Patent Application
6. ☒ Interview Summary (PTO-413),
Paper No./Mail Date 20100616
7. ☒ Examiner's Amendment/Comment
8. ☒ Examiner's Statement of Reasons for Allowance
9. ☐ Other ____.

Examiner's Amendment and Statement of Reasons for Allowance

1. This action is responsive to Applicant's amendment filed on March 18, 2010.

Information Disclosure Statement

2. The contents of information disclosure statement filed 06/15/2010 which are lined through fail to comply with the provisions of 37 CFR 1.98 because these contents cannot be listed in a printing patent. These contents which are the internal office actions replying by patent Examiners should not be listed in a printing patent.

In accordance to 37 CFR 1.98 (a) (2) (ii) or (iv), the considered portions will be listed:

- (ii) Each publication or that portion which caused it to be listed, other than U.S. patents and U.S. patent application publications unless required by the Office;
- (iv) All other information or that portion which caused it to be listed.

Thus, every content submitted under 37 CFR 1.98 when being **initialed** will cause to be listed in a printing patent. Therefore, pursuant to 37 CFR 1.98, Applicants should submit the contents as U.S. patent application publications and/or pending unpublished U.S. applications. An office action cannot be the portion which is caused to be listed in accordance to 37 CFR 1.98 (a) (2) (ii) or (iv).

As per request for considering the portions/contents that are the internal Examiner office actions, these portions/contents are considered by the Examiner, but they are lined through because of 37 CFR 1.98 (a) (2) (ii) or (iv).

Examiner's Amendment

3. An examiner's amendment to the record appears below. Should the changes and/or additions be unacceptable to applicant, an amendment may be filed as provided by 37 CFR 1.312. To ensure consideration of such an amendment, it **MUST** be submitted no later than the payment of the issue fee.

Authorization for this examiner's amendment was given in a telephone interview with Mr. Caleb Pollack, Registration Number 37,912, and Ms. Tami Ben Ari on June 15, 2010 for obviating any potential 101 issues and put the claims in condition for allowance.

The application has been amended as follows:

1. (Currently Amended) A method comprising:
defining, by a language builder module executed by a processor of a language builder station, a domain-specific language usable in defining a modeling environment and having a dynamic component and a static component, wherein behavior definitions of said static component are modifiable by said dynamic component; and
applying on a modeling environment, by a language runtime module executed by a processor of a modeling station, modification by said dynamic component of behavior definitions of said static component, so as to change definitions of modeling process, wherein said dynamic component is a domain-

specific language definition applied on said static component; and the domain-specific language definition is built via a DSL builder station.

15. (Currently Amended) The method of claim 14, wherein creating comprises:
generating a recommended modeling process to be used during creation of said one or more elements of said model in accordance with a mentor modeling definition of said domain-specific language wherein the modeling definition may include at least one of a list consisting of : modeling process, modeling activities, modeling transitions, modeling artifacts, progress criteria, process-sensitive help items, step-by-step wizards, next activity prompts, estimation of modeling progress, working point sensitive helps, checking model status, detecting conflicts, providing warning message about detected conflicts, suggesting activities to be performed, estimating percentage of tasks executed, phase-sensitive helps and methodology guiding.

19. (Currently Amended) A system for accelerated modeling, the system comprising:

a language builder station running on a processor for executing a language builder module to define a domain-specific language usable in defining a modeling environment and having a dynamic component and a static component, wherein behavior definitions of said static component are modifiable by said dynamic component said dynamic component; and

a modeling station running on a processor for executing a language runtime module to apply on a modeling environment modification by said dynamic component of behavior definitions of said static component so as to change

definitions of modeling process, wherein said dynamic components is a domain-specific language definition applied on said static component, . wherein said dynamic component is a domain-specific language definition applied on said static component; and the domain-specific language definition is built via a DSL builder station

36. (Currently Amended) A non-transitory machine-readable medium having stored thereon instructions that, when executed by a machine, result in:

defining a domain-specific language usable in defining a modeling environment and having a dynamic component and a static component, wherein behavior definitions of said static component are modifiable by said dynamic component; and

applying on a modeling environment modification by said dynamic component of behavior definitions of said static component so as to change definitions of modeling process, wherein said dynamic component is a domain-specific language definition applied on said static component, wherein said dynamic component is a domain-specific language definition applied on said static component; and the domain-specific language definition is built via a DSL builder station.

-- The End --

Examiner's Statement of Reason(s) for Allowance

4. Claims 1-39 are allowed.

5. The following is an examiner's statement of reasons for allowance:

The prior arts of record: **Freed** et al., teaches a software modeling environment is presented that supports the development and execution of software that can be dynamically configured. This is achieved by introducing a design-time object modeling construct called a dynamically configurable type (DCT) that supports the subdivision of an object into a plurality of subsets of semantics and allows dynamic merging of these semantic subsets together in different combinations in order to allow the resultant object to exhibit different semantics over time to serve different purposes. Specific process modeling constructs are included which take advantage of this semantic merging capability. Also included are other supporting design-time and run-time constructs and services. **Makhlouf**, teaches a physics-like computational models called M models. These models enable the specification and design automation of an integrated architecture of a real system. A new class of adaptive systems engineering tools called Adaptive Model-Reference (AMR) tools, supports the M computational model. AMR tools are used to visually specify the integrated architecture model of a real system, assess it's value, and automate the static and dynamic binding of it's model components into adaptive systems. These systems can be adapted during system development to compensate for requirement changes and design errors, and during run time operation to compensate for unanticipated operational system conditions. AMR tools enable the verification and validation of adaptive real system designs built in compliance with a declared enterprise wide technical architecture. Architecture components can be

specified using AMR tools or can be imported into the AMR tool set. **Coulthard et al.**, teaches a method and apparatus that has the ability to convert display pages of interactive legacy applications for use on a network, such as the Internet. The display source code of the legacy application is first parsed into a network-based language, such as XML, preserving the structure and hierarchy of the display source to create a plurality of network user interface pages. The network user interface pages are then converted to a dynamic platform-independent language in which the static portion of the display page is converted to a web page, such as a **JavaServer™** Page, and the dynamic portion of the display page for input/output/feedback is converted to data objects, such as **JavaBeans**. The intermediate network user interface pages may be stored on the server with the legacy application. The conversion program to convert the intermediate network user interface pages may be stored on a computer program product, a client connected to the server with the legacy application, or the server itself. The method of this invention provides an environment such that when an application is invoked from a client, the runtime data manager may use either the traditional display source code or the generated intermediate network user interface pages as its user interfaces. **Ben-Romdhane et al.**, teaches a system and method for manipulating software is provided. The system analyzes a body of source code and derives a set of components from the software architecture inherent in the body of source code. The system further determines the internal and external functional and data dependencies of each component and the control flow of the software and creates an information model that represents the components, their dependencies, and the control flow of the program. Additionally, the system provides a graphical interface that presents the information model and allows the macro and micro

structure of the model to be viewed and modified. Furthermore, components can be combined and separated to refine the software architecture represented by the information model and documentation and new source code can be provided for inclusion in the information model. **Konstantinou** et al., teaches methods and systems for autonomously managing computer networks are provided. In a preferred embodiment of the invention, management functions are organized in a novel two-layer peer-to-peer (P2P) architecture. The bottom layer organizes management information in a unified object-relationship model, that is instantiated in a distributed transactional object repository. The top layer unifies the traditional roles of managers and elements into a single autonomic management peering layer. Autonomic elements use the repository as a primary management repository, and effect autonomic behavior in terms of transactions over the shared model state. Hunt, teaches a computer readable medium contains a data structure, which, in some embodiments, is an interface wrapper. The interface wrapper stores information about the interface such as a first data field that contains a reference to instrumentation, a second data field that contains data which represents a reference to an interface of a software program, and a third data field which contains data representing an identity of the unit of software. When a client unit calls a member function of the interface, the interface wrapper intercepts the call and invokes the instrumentation. **Katzeff**, teaches a synthesizer means for generating software for a computer which is programmed for controlling a physical system. The software generated by the synthesizer represents a new function to be incorporated in the existing system. The synthesizer includes a device for receiving a formal description representative of the new function in a specification language and for translating the specification into a base document. The base document is further

processed by document processing devices for handling the static, interface and dynamic parts of the description to produce an error-free base document. The complete base document is translated by an information processing device into an internal code document which is used by a check device and a simulation device. A compiling device translates the internal code document into an intermediate code document suitable for input to said computer. **Toutonghi** et al., teaches dynamic mapping from an ActiveX component model to a JavaBean™ model. An ActiveX compatible object is created at run-time for those JavaBean™ components that an ActiveX client application wishes to utilize. In another embodiment, upon a call to CoGetObject or related methods using class identifiers, a COM-callable wrapper is generated and a mapping table containing COM dispatch identifiers is scanned to ensure the dispatch identifiers are unique. If they are not, the dispatch identifiers associated with JavaBean™ component interfaces are adjusted to make them unique. Hanna et al., teaches a software system is defined by a tree of system models which are written in a functional language. During a build of the software system, the functions are interpreted and the results of the expensive expressions are cached. Each function is examined before interpretation to see if it has been evaluated before. If a function has already been evaluated, the cached result is retrieved by the evaluator and the time which would have been spent re-evaluating the function is saved. Smith et al., teaches methods, systems, and computer program products for identifying computer source code constructs are disclosed. According to one method, computer source code is converted to a format suitable for an automated inference engine. The automated inference engine receives as inputs the converted source code, a set of elemental design patterns defining patterns to be identified, and a set of rules defining relationships between patterns.

The automated inference engine outputs proofs indicative of patterns present in the source code. The proofs may be converted to a source code pattern report.

Bhargava et al., teaches a method for monitoring, predicting performance of, and managing Business Operations by the simultaneous, real-time Integration, Normalization and Correlation of direct measurements at the Business Layer and other Layers of Business Operations. The other Layers considered, may include, for example, Application and Infrastructure Layers. The system enables the user to automate sophisticated management tasks by Correlating measurements of activity, performance and availability at all Layers of Business Operations. Significantly, the techniques described herein extend the domain of the Correlations across real-time measurements from all Layers of Business Operations, giving central importance to the measurements in the Business Layer within the Correlations.

New arts made of record: US 2001/0045963 by **Marcos** et al., teaches a graphical user interface (GUI) and accompanying functionality for binding Web page definitional elements to a back-end state (e.g., client- or server-side back-end state) and custom logic is provided. In one embodiment, a template containing definitional elements, custom logic, and bindings are generated that define all or a portion of a Web page based on input received and functionality provided by the invention. US Patent 6,496,870 B1, by **Faustini**, teaches system for collaborating components or objects in a visual development environment is detailed.

Collaboration is effected by augmenting eligible components or objects with appropriate collaboration code and registering such components or objects with a server application designated for that purpose which resides on the same HTTP server where the applet that spawned the components to be collaborated also resides. The server application first registers objects or components or portions

thereof to be collaborated, builds a record of such links and thereafter interacts with the collaborated components or designated portions thereof to publish, unpublish or update those components and objects, or portions thereof, in accordance with the application server record. US 2004/00260528 A1 by **Ballagh** et al., teaches method and apparatus for interfacing a high-level modeling system (HLMS) with a reconfigurable hardware platform for co-simulation. In one embodiment a boundary-scan interface is coupled to the HLMS and is configured to translate HLMS-issued commands to signals generally compliant with a boundary-scan protocol, and translate signals generally compliant with a boundary-scan protocol to data compatible with the HLMS. A translator and a wrapper are implemented for configuration of the hardware platform. The translator translates between signals that generally compliant with the boundary-scan protocol and signals that are compliant with a second protocol. US 2002/0118220 A1 by **Lui**, teaches a Cooperative Help Assistance (CHA) system and method provide real-time user assistance for one or more windows-based Graphic User Interface (GUI) applications or a single application's different subsections such as web pages, running concurrently in any operating system. The CHA System enables the development of an informative assistance object independently from the original source code or development environment of the target Host Application. The assistance object can be selected by any number of user interfaces from sophisticated inference driven interactive interface search tools or categorized lists. By intercepting and monitoring user actions on a Host Application, the CHA system performs intelligent assistance in the context of the target host application program. Utilizing a Host Application Model, the CHA System and method dynamically assemble many elements in real-time or just-in-time to produce

assistance sequences or elements very efficiently without having to code every interface path permutation. Paths can be dynamically generated from the Host Application Model, which enables a real-time module to offer intelligent, contextual assistance as well as real-time construction of automated, accelerated CHA Sequences or Procedures that require little or no user interaction. All assistance and information are processed and expressed by an extensive multitasking, multimedia subsystem for two dimensional (2D) and real-time three-dimensional (3D) application interfaces, which greatly enhances and extends the effectiveness of any explanation or material expression. However, none of them, taken alone or in combination, teaches the features in such a manner as recited in independent claims 1, 14, and 27.

Any comments considered necessary by applicant must be submitted no later than the payment of the issue fee and, to avoid processing delays, should preferably accompany the issue fee. Such submissions should be clearly labeled "Comments on Statement of Reasons for Allowance."

6. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Chih-Ching Chow whose telephone number is 571-272-3693. The examiner can normally be reached on 8:00am - 4:30pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Wei Zhen can be reached on 571-272-3708. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

/Chih-Ching Chow/

Examiner, Art Unit 2191

06/17/10

/Wei Y Zhen/

Supervisory Patent Examiner, Art Unit 2191